β3

-- This Application has been assigned serial number 09/550,191 and a filing date of April 17, 2000. --

REMARKS

This Amendment is submitted in response to the Office Action mailed on January 30, 2003. Claims 1 - 7 are pending, and all stand rejected at present.

Claims 8 - 13 were added. Support is found in the Specification, at least beginning on page 31, section entitled "One Form of Invention." For example, the last sentence of the first paragraph in this section refers to modifying software in a payment switch, and installing the modified software.

The Specification has been amended to further identify a patent application which was incorporated by reference.

Response to 112 - Rejections

Claim 1

Claim 1 has been amended.

In response to the rejection of claim 1(b)(iii), based on the term "at least one," Applicant requests a citation of authority in support of the rejection.

One reason is that the reason given by the Office Action is that the claim language is "open ended." However, being "open

ended" does not, by itself, render a claim indefinite. Indefiniteness is the basis of any 112 - rejection. That is, "at least one" covers all numbers from one to infinity. There is nothing indefinite about that. It may be a large collection of numbers, but it is, in fact, definite. ~ *[handwritten annotation]*

If the rejection is, in effect, asking "how many modules EXACTLY are being claimed ?" Applicant points out that an exact number need not be specified. The answer is "at least one." *[handwritten annotation]*

Further, it is axiomatic that the claims are read as-a-whole. The phrase "at least one" is read with the rest of the claim, not in isolation.

Further still, the undersigned attorney did a search of the PTO's database, looking for the phrase "at least one" in patents issued from 1976 to the present. Almost one million hits were returned: 919,632 hits, to be precise.

Applicant fails to see how this rejection can be justified, in view of the fact that, since 1976, almost one million patents have used the claim language in question.

Claims 6 and 7

Applicant respectfully submits that the PTO mis-applies the law of claim interpretation.

The question which the claim must answer is **NOT** this: "Which specific ones of the PAK_MOD modules contain unit B, with no unit

C ?"

Rather, the question is "Do **SOME** of the PAK_MOD modules contain unit B, with no unit C ?" If so, then the claim recitation (6(b)(iii)(B) in this case) is found.

From another point of view, Applicant **does not know** how an infringer will choose to arrange the modules. For that reason alone, Applicant cannot say "which" modules will be covered by the claim language.

RESPONSE TO 102 - REJECTIONS

Claim 1

Claim 1 recites:

> 1. A method of constructing a plurality of software systems, comprising the following steps:
>
> a) maintaining an inventory of software modules, which includes:
>
> > i) a group of type A modules; and
>
> > ii) a collection of type B modules;
>
> b) when constructing each software system,
>
> > i) including copies of the entire group of type A modules;
>
> > ii) including copies of [selected] <u>some or all</u> type B modules; and
>
> > iii) generating at least one customized module, which is a copy of neither a type A nor a Type B module.

6

<u>Claim 1(b)(i) and (ii)</u>

The Office Action relies on Yates, column 18, lines 1 - 13 to show claim 1(b)(i) and (b)(ii). Those claim passages state that

1)   all of the type A modules are included in

the software system being constructed

and

2)   some or all of the type B modules are

included.

That passage of Yates relied on by the PTO is here set forth:

> Known   constructions,   where   policies[1]   are embedded in the objects, require rewriting of code in the object to change behavior.
>
> External policies allow not only changes in behavior to be achieved more easily but also more freely, and can allow extra behaviors (which are composed from combinations/permutations of a programmed set of operations) to be performed even if these were not originally anticipated.
>
> The concept of policies is such that an object must have access to a "Policy Interpreter."
>
> This can be internal or external to the object.
>
> In order to locate policies, a policy server might be provided, again either internal or external to an object.

---

[1]   "Policy" is apparently jargon for a section of computer code.

(Yates, column 18, lines 1 - 13.)

Plainly, the Yates passage has no relevance to claim 1(b)(i) and (b)(ii). This passage fails to show anything corresponding to type A and type B modules, nor the particular selections recited in claim 1(b)(i) and (ii).

MPEP § 2131 states:

> A claim is anticipated only if **each and every element** as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.

Applicant submits that claim 1(b)(i) and (b)(ii) have not been shown in Yates, and thus claim 1 cannot be anticipated.

In addition, Applicant points out that it is axiomatic that, in order to **anticipate** claim 1, the Yates reference must **infringe** claim 1. (See PATENTS, A Treatise on the Law of Patentability, Validity, and Infringement, by D. Chisum, section 3.02[1], entitled, "The Classic Infringement Test.")

Applicant submits that the Yates passage set out above clearly does not show claim 1(b)(i) and (b)(ii).

Therefore, Applicant submits that Yates does not anticipate claim 1.

Claim 1(a)

Claim 1(a) recites maintaining type A and type B modules. The

Office Action relies on identical passages in Yates to show both of these module types. Those passages are

        1)   The Abstract

        2)   Column 2, lines 57 - 65,

        3)   Column 4, lines 3 - 12,

        4)   Column 5, lines 40 - 55, and

        5)   Column 18, lines 1 - 13.

As to item (1), Yates' Abstract merely refers to selecting "reusable software modules." There is no reference to anything analogous to types A and B.

As to item (2), the passage merely refers to selecting a "set" of software modules.

Item (3) merely refers to "adding" or "modifying" software modules.

Item (4) refers to changing a set of software modules which is used at a given time.

Item (5) refers to the Yates passage which was quoted above. That passage refers to modifying a "set" of software modules.

Therefore, Applicant submits that the two "types" of module recited in claim 1(a) have not been shown in Yates.

Claim 1(b)(iii)

To show claim 1(b)(iii), the Office Action relies on the five identical passages used to reject claims 1(b)(i) and (ii). Those

passages, summarized immediately above, fail to show

> iii)   generating at least one customized
> module, which is a copy of neither a type
> A nor a Type B module,

as recited in claim 1(b)(iii).

Claims 2 - 5

Claims 2 - 5 depend from claim 1, and are considered allowable based on claim 1.

Claim 6

Claim 6(a)

Applicant points out that claim 6(a) recites:

> a)   fabricating a collection of software
> systems, each of which contains

and then lists four types of module.

Restated, claim 6(a) states that every "software system" in the "collection" contains (at least) the four modules listed in 6(a)(i) through (a)(iv).

The Office Action cites two passages in Yates to show this, namely,

> 1)   column 2, lines 38 - 65
>
> and

2)   column 4, lines 13 - 65.

However, those passages contain nothing more than generalized statements indicating that, in different situations, systems may be designed which are different.

That is **directly contrary** to the claim recitations in question. One reason is that claim 6(a) states that every software system contains four specific modules. Thus, in that respect, every software system is **identical**.

That is contrary to the cited passages in Yates.

Claim 6(b)

Claim 6(b) states that all of the "software systems" of claim 6(a) will contain two elements, namely (1) identical CONTROL modules and (2) identical COM_MOD modules. The Office Action relies on two passages in Yates to show this.

One passage is column 4, lines 26 -35. However, that passage merely states that "at least one . . . software agent" is equipped with certain functionality. That does not show the claim recitations in question, and is actually inconsistent with it.

It is inconsistent because the Yates-passage only focuses on **ONE** software agent, and lists some properties of that agent. It fails to identify a **GROUP** of agents. Claim 6 refers to properties of a "collection of software systems." Yates' discussion of a **SINGLE** software agent does not show the properties of a **GROUP** as

in claim 6.

The other passage relied on by the PTO is Yates column 18, lines 1 - 13. That passage is set out <u>verbatim</u> above, and clearly does not show claim 6(b)(i) and (b)(ii).

Claim 6(b)(iii)

Applicant points to claim 6(b)(iii), which is repeated here:

> iii) fabricating PAK_MOD modules in all systems, such that:
>
>> A) copies of a software unit A is contained in every PAK_MOD module;
>>
>> B) some PAK_MOD modules contain a software unit B with no unit C; and
>>
>> C) some PAK_MOD modules contain a software unit C with no unit B.

The undersigned attorney has examined the passages in Yates which are cited to show these recitations, and cannot locate the recitations in those passages. [Actually, the passages used by the PTO are the same as used for claim 6(b)(i) and (b)(ii).]

Further Consideration of Claim 6

Claim 6(a)(iii) refers to a "communications module (COM_MOD) which accepts and delivers message packets." Yates, column 14, line 49 et seq., refers to a Manager which uses data packets for

file transfer. It is Assumed _arguendo_ that Yates' Manager shows the recited COM_MOD.

Claim 6(b)(ii) states that, during fabrication of software systems, "fabricating identical COM_MOD modules in all systems." The Office Action relies on two passages in Yates to show this.

One passage is column 4, lines 26 - 35. However, that passage, in essence, states that each "software agent" is "customized" for a specific purpose. Thus, the agents will be different. That does not state, or even imply, "identical COM_MOD modules in all" agents.

The second passage is column 18, lines 1 - 13. That passage was set forth _verbatim_ above. That passage merely refers to a process of modifying software. That does not state, or even imply, "identical COM_MOD modules in all" agents.

In fact, it would tend to show the opposite. If software (ie, the COM_MOD modules) is modified, then for "identical COM_MOD modules" to exist in all agents, all those modules must be modified in **the same manner**. That has not been shown in Yates.

General Observations on Yates

If an attempt is made to apply claim 1 to Yates, the undersigned attorney believes that the only possible elements in Yates which apply to (1) the "software system," (2) the type A modules, and (3) the type B modules of claim 1 are the following,

respectively:

        --   The "agents" (corresponding to the "software system"),

        --   The SIBBs, Service Independent Building Blocks (corresponding to the type A modules), and

        --   The "adaptors" (corresponding to the type B modules.)

(See column 17, lines 12 - 22.)

However, several problems arise. Claim 1(b)(i) states that "the entire group" of the type A modules is included in the "software system." Yates expressly states that is not so. He states that the SIBBs in the "agents" change over time. (Column 18, lines 42 - 48.) Thus, as a minimum, the "entire group" of the SIBBs does not remain constant.

An the undersigned attorney can find no statement that "the entire group" of the SIBBs is given to each "agent" in the first place.

Another problem arises in connection with claim 3, which states:

> 3. Method according to claim 2, wherein functions (3) and (4) are performed using type A modules exclusively.

"Functions (3) and (4)" of claim 2 refer to transfer of messages

into, and out of, the system. Yates' SIBBs do not do that. His Communications Session Manager does that. (Column 14, line 49 et seq.) That Manager does not appear to be a SIBB.

Therefore, Applicant requests that the "group of type A modules" be identified in Yates.

Response to RESPONSE TO ARGUMENTS

As to the Yates passage, the passage merely refers to modifying computer code. The claim language in question does not recite that.

The language "at least" is accurate: the claim is a "comprising" type of claim. Other elements can be present in an infringing device. Further, it appears that the Office Action is demanding that Applicant submit an amendment to an amendment. That is not allowed.

As to In re Haza, even if the rule stated by the Office Action be correct, that rule only applies when the claim elements in question have been shown in the prior art. That has not been done here.

Further, the stated rule is that "a plurality of elements has no patentable significance unless a new and unexpected result is produced." That statement is necessarily incorrect. It is contrary to section 102, which states that a "novel" combination of old elements is patentable (if non-obvious, of course).
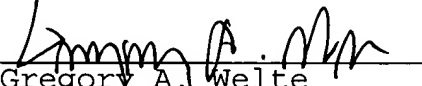
15

Further still, the "rule" is inapplicable here. All claims are method claims.

Conclusion

Applicant requests that the rejections to the claims be reconsidered and withdrawn.

Applicant expresses thanks to the Examiner for the careful consideration given to this case.

Respectfully   submitted,

Gregory A. Welte
Reg. No. 30,434

NCR Corporation
1700 South Patterson Blvd.
WHQ - 5E
Dayton, OH 45479
May 30, 2003
(937) 445 - 4956
ATTACHMENT: Annotated Claim(s) Showing Amendments

ATTACHMENT: Annotated Claim(s) Showing Amendments

1.   A method of constructing a plurality of software systems, comprising the following steps:

   a)   maintaining an inventory of software modules, which includes:

   i)   a group of type A modules; and

   ii)   a collection of type B modules;

   b)   when constructing each software system,

        i)   including copies of the entire group of type A modules;

        ii)   including copies of [selected] some or all type B modules;

        and

        iii)   generating at least one customized module, which is a copy of neither a type A nor a Type B module.